
Article

[Dmitry Maslennikov](#) · Feb 6, 2023 4m read

[Open Exchange](#)

Welcome irissqlcli - advanced terminal for IRIS SQL

Let me introduce my new project, which is [irissqlcli](#), REPL (Read-Eval-Print Loop) for InterSystems IRIS SQL

- Syntax Highlighting
- Suggestions (tables, functions)
- 20+ output formats
- stdin support
- Output to files

Install it with pip

```
pip install irissqlcli
```

Or run with docker

```
docker run -it caretdev/irissqlcli irissqlcli iris://_SYSTEM:SYS@host.docker.internal:1972/USER
```

Connect to IRIS

```
$ irissqlcli iris://_SYSTEM@localhost:1972/USER -W
Password for _SYSTEM:
Server: InterSystems IRIS Version 2022.3.0.606 xDBC Protocol Version 65
Version: 0.1.0
[SQL]_SYSTEM@localhost:USER> select $ZVERSION
+-----+
+-----+
| Expression_1
|
+-----+
+-----+
| IRIS for UNIX (Ubuntu Server LTS for ARM64 Containers) 2022.3 (Build 606
U) Mon Jan 30 2023 09:05:12 EST |
+-----+
+-----+
1 row in set
Time: 0.063s
[SQL]_SYSTEM@localhost:USER> help
+-----+-----+-----+
+-----+
| Command | Shortcut | Description
|
+-----+-----+-----+
+-----+
| .exit   | \q      |
|         |         | Exit.
```

```

| .mode      | \T                | Change the table format used to output results.
|           |                   |
| .once      | \o [-o] filename | Append next result to an output file (overwrite using -o).
| .schemas   | \ds              | List schemas.
|           |                   |
| .tables    | \dt [schema]     | List tables.
|           |                   |
| \e         | \e              |
|           | Edit command with editor (uses $EDITOR).
| help       | \?              | Show this help.
|           |                   |
| nopager    | \n              | Disable pager, print to stdout.
| notee      | notee           | Stop writing results to an output file.
|           |                   |
| pager      | \P [command]     | Set PAGER. Print the query results via PAGER.
| prompt     | \R              | Change prompt format.
| quit       | \q              | Quit
|           |                   |
| tee        | tee [-o] filename | Append all results to an output file (overwrite using -o).
+-----+-----+-----+-----+
-----+
Time: 0.012s
[SQL]_SYSTEM@localhost:USER>

```

```
$ irissqlcli --help
```

```
Usage: irissqlcli [OPTIONS] [URI] [USERNAME]
```

Options:

```

-h, --host TEXT      Host address of the IRIS instance.
-p, --port INTEGER   Port number at which the IRIS instance is listening.
-U, --username TEXT   Username to connect to the IRIS instance.
-u, --user TEXT       Username to connect to the IRIS instance.
-W, --password        Force password prompt.
-v, --version         Version of irissqlcli.
-n, --namespace TEXT  namespace name to connect to.
-q, --quiet           Quiet mode, skip intro on startup and goodbye on exit.

-l, --logfile FILENAME Log every query and its results to a file.
--irissqlclirc FILE    Location of irissqlclirc file.
--auto-vertical-output Automatically switch to vertical output mode if the result is wider than the terminal width.
--row-limit INTEGER    Set threshold for row limit prompt. Use 0 to disable prompt.
-t, --table           Display batch output in table format.
--csv                 Display batch output in CSV format.
--warn / --no-warn    Warn before running a destructive query.
-e, --execute TEXT     Execute command and quit.
--help                Show this message and exit.

```

or in Python Embedded mode (requires %ServiceCallIn enabled)

```
$ irissqlcli iris+emb:///USER
```

```
Server: IRIS for UNIX (Ubuntu Server LTS for ARM64 Containers) 2022.2 (Build 368U
```

```
) Fri Oct 21 2022 16:39:41 EDT
Version: 0.1.0
[SQL]irisowner@/usr/irissys/:USER>
```

It supports stdin, so you can pipe some SQL file with a bunch of SQL queries and irissqlcli commands. For instance, this command, will produce 3 files in different formats (out of more than 20 formats available)

```
$ cat <<EOF | irissqlcli iris://_SYSTEM:SYS@localhost:1972/USER
```

```
.mode csv;
tee -o test.csv;
select top 10 TABLE_SCHEMA, TABLE_NAME
from information_schema.tables
order by TABLE_SCHEMA, TABLE_NAME;
notee;
```

```
.mode latex;
tee -o test.tex;
select top 10 TABLE_SCHEMA, TABLE_NAME
from information_schema.tables
order by TABLE_SCHEMA, TABLE_NAME;
notee;
```

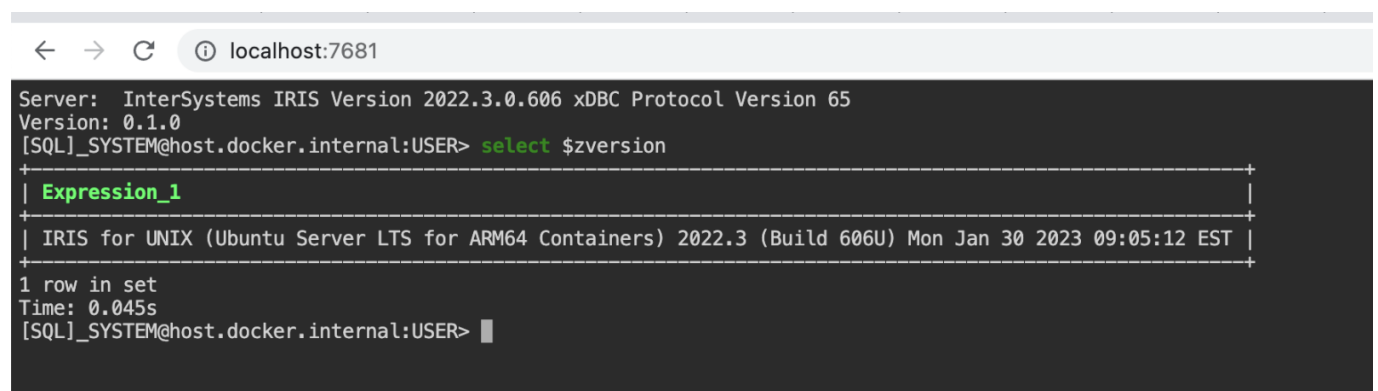
```
.mode html;
tee -o test.html;
select top 10 TABLE_SCHEMA, TABLE_NAME
from information_schema.tables
order by TABLE_SCHEMA, TABLE_NAME;
notee;
```

```
EOF
```

Additionally it's possible to run a web terminal with docker

```
docker run -d --name irissqlcli \
  --restart always \
  -p 7681:7681\
  caretdev/irissqlcli-
web irissqlcli iris://_SYSTEM:SYS@host.docker.internal:1972/USER
```

<http://localhost:7681/>



And with docker-compose

```
version: '3'
services:
  iris:
    image: intersystemsdc/iris-community
    ports:
      - 1972
      - 52773
    command:
      - -a
      - '##class(Security.Users).UnExpireUserPasswords("*")'
  cli:
    image: caretdev/irissqlcli-web
    ports:
      - 7681:7681
    environment:
      - IRIS_HOSTNAME:iris
      - IRIS_PORT=1972
      - IRIS_NAMESPACE=USER
      - IRIS_USERNAME=_SYSTEM
      - IRIS_PASSWORD=SYS
```

Please [vote for the project](#) in the contest

[#Best Practices](#) [#Embedded Python](#) [#Python](#) [#SQL](#) [#Tools](#) [#InterSystems IRIS](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/welcome-irissqlcli-advanced-terminal-iris-sql>